Characterizing Biological Networks Using Subgraph Counting and Enumeration

George Slota Kamesh Madduri

madduri@cse.psu.edu

Computer Science and Engineering

The Pennsylvania State University

SIAM PP14

February 20, 2014

# of speakers mentioning protein interaction networks

4 out of 5 (including me)

# of speakers mentioning BFS

3 out of 5

Acknowledgments

- NSF ACI award 1253881
- NSF XSEDE allocation TG-CCR13006
- Siva Rajamanickam, Sandia

- Current and past organizers of parallel graph analysis minisymposia at SIAM PP

CONFERENCES >

Archives and Future Meetings

By Subject | By Year | Lectures

Parallel Processing

- SIAM Conference on Parallel Processing for Scientific Computing (PP14)
- SIAM Conference on Parallel Processing for Scientific Computing (PP12)
- SIAM Conference on Parallel Processing for Scientific Computing (PP10)
- SIAM Conference on Parallel Processing for Scientific Computing (PP08)
- SIAM Conference on Parallel Processing for Scientific Computing (PP06)
- SIAM Conference on Parallel Processing for Scientific Computing (PP04)
- Tenth SIAM Conference on Parallel Processing for Scientific Computing (SIAG/SC) (PP01)
- Ninth SIAM Conference on Parallel Processing for Scientific Computing (PP99)
- Eighth SIAM Conference on Parallel Processing (PP97)
- Seventh SIAM Conference on Parallel Processing (PP95)
- Sixth SIAM Conference on Parallel Processing (PP93)
- Fifth SIAM Conference on Parallel Processing (PP91)
- Fourth SIAM Conference on Parallel Processing (PP89)
- Third SIAM Conference on Parallel Processing (PP87)
- Second SIAM Conference on Parallel Processing (PP85)

SIAM PP06

MS16
Parallel Graph Algorithms

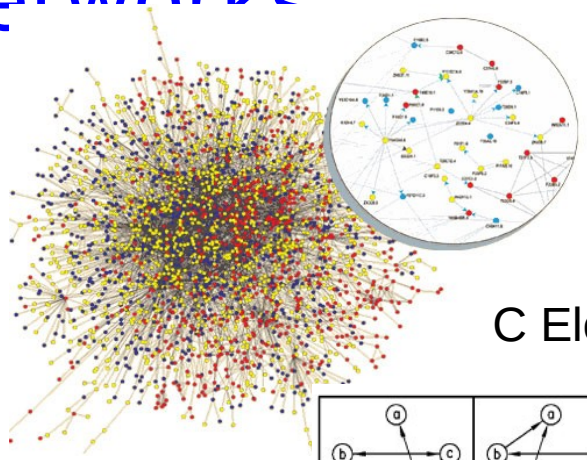
4:00 PM - 6:00 PM
Room: Carmel - 2nd Floor

Graph algorithms have long played a pivotal enabling role in many applications of parallel computing, su
applications. While these are still active research areas, important, emerging areas of science, which are fi
computational biology, scientific data mining, and network analysis, are changing the relationship betwee
performance, graph algorithms are now customers for parallel computing. This minisymposium will addr
approaches.

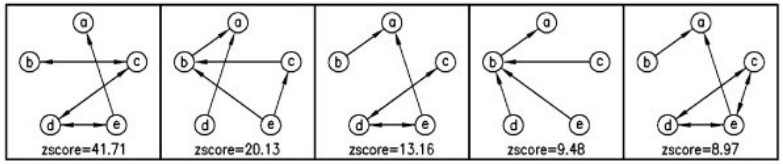
Organizer: Bruce Hendrickson
Sandia National Laboratories, USA
Ali Pinar
Sandia National Laboratories, USA

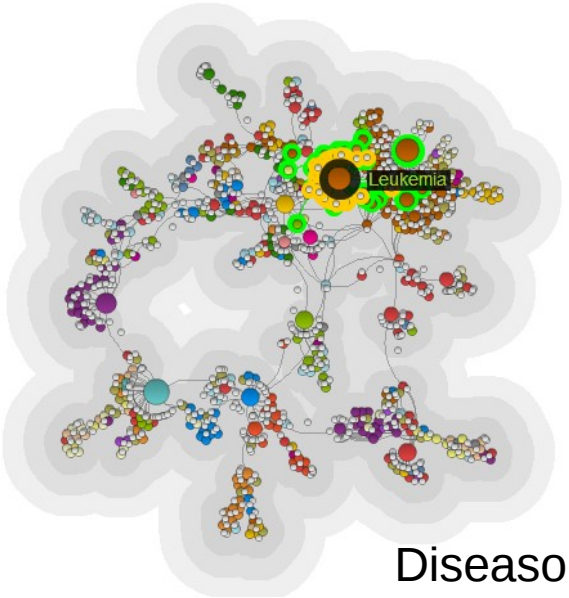
4:00-4:25 Computing Approximate Matchings in Parallel abstract
Fredrik Manne,
4:30-4:55 Scalable Graph-Theoretical Approaches to Biological Network Analysis abstract
Nagiza F. Samatova, ; Yun Zhang, ; Henry Suters, ; Faisal Abu-Khzam, ; Michael A. Langston,
5:00-5:25 The Parallel Boost Graph Library abstract
Andrew Lumsdaine and Douglas Gregor,
5:30-5:55 A Graph Infrastructure for Multithreaded Architectures abstract
Jonathan Berry,

Network motifs in protein interaction networks (PINs) and gene-disease networks



C Elegans PIN



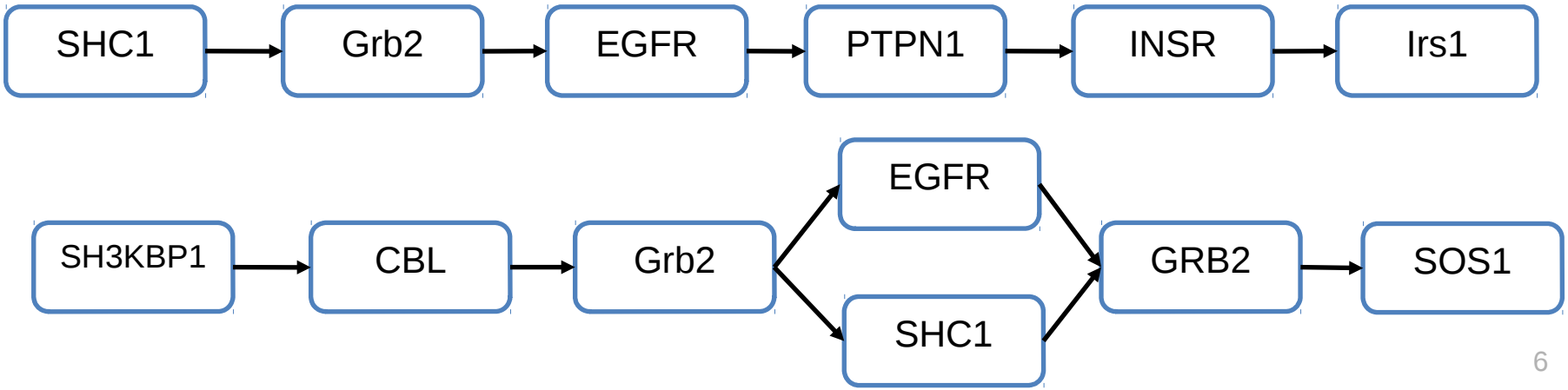


Diseasome

DisGeNET

Signaling pathways in PINs

Two high-scoring pathways from human PIN



Our new parallel approach: FASCIA

- For subgraph counting: Parallel and memory-efficient implementation of an approximation algorithm based on the color-coding technique
  - $O(2^k e^k m)$ work (exhaustive search requires $O(n^k)$ work)
- Significantly faster (at least 10X) than prior parallel color-coding implementations

- Multicore parallelism: Slota and Madduri, Proc. ICPP 2013
- Distributed-memory parallelism and network analysis: Slota and Madduri, Proc. IPDPS 2014 (to appear)

analysis: Slota and Madduri, Proc. IPDPS 2014 (to appear)

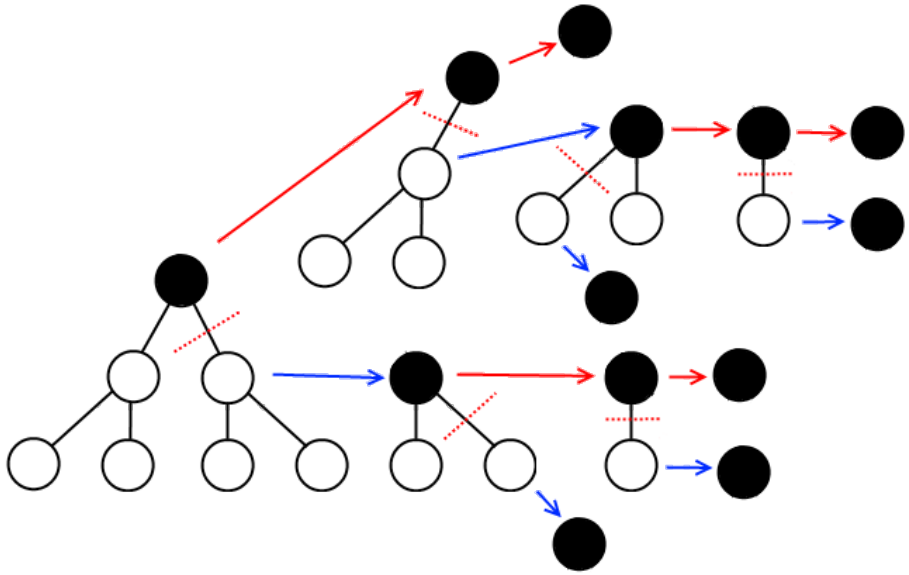
FASCIA features

- Get approximate counts of tree-structured templates
- Determine network motifs
- Comparative network analysis using
  - Graphlet frequency distances (GFD)
  - Graphlet degree distributions (GDD)
  - Graphlet degree signatures (GDS)
- Cluster networks into categories
- New: simple modifications to find signaling pathways in protein interaction networks (PINs)

This talk: FASCIA applied to PIN analysis and comparative network analysis

- The dynamic programming scheme and shared-memory parallelism
- FASCIA and approximate counting for finding motifs in protein interaction networks
- Comparative network analysis
- Parallel performance results

FASCIA Algorithm and Parallelization Overview



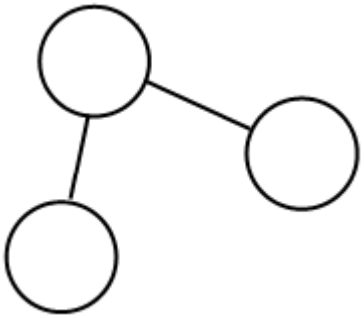
- Template partitioning
- Count number of colorful occurrences of template
  - Memory-intensive step
  - $O(m2^k)$ work; $O(n2^k)$ space requirements
  - Optimizations, Parallelization of dynamic programming-based counting step
- Estimate the total number of occurrences

occurrences

Color-coding approximation strategy

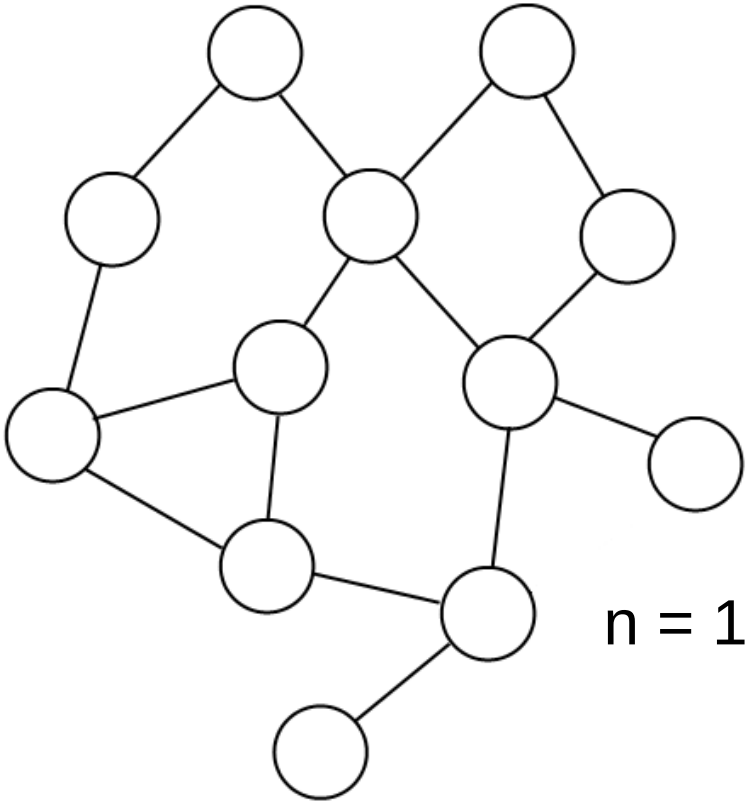
- Alon et al., 1995: approximate counting of tree-like non-induced subgraphs

Template



k = 3

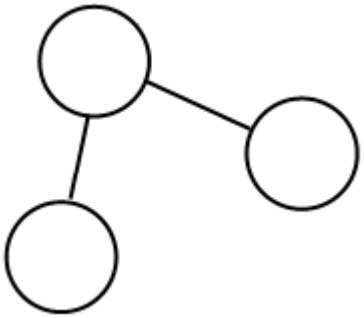
Larger graph



n = 12, m = 15

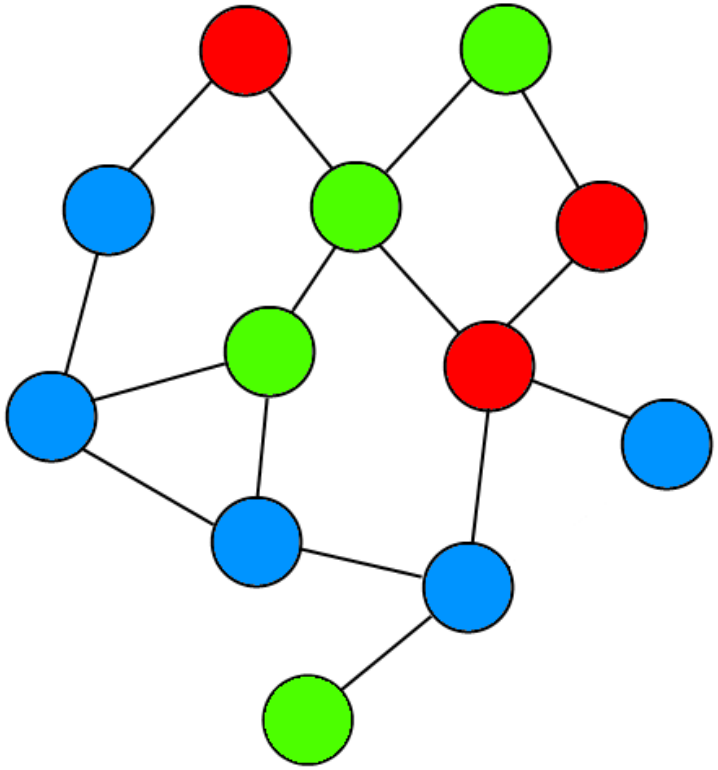
Color-coding approximation strategy

Template



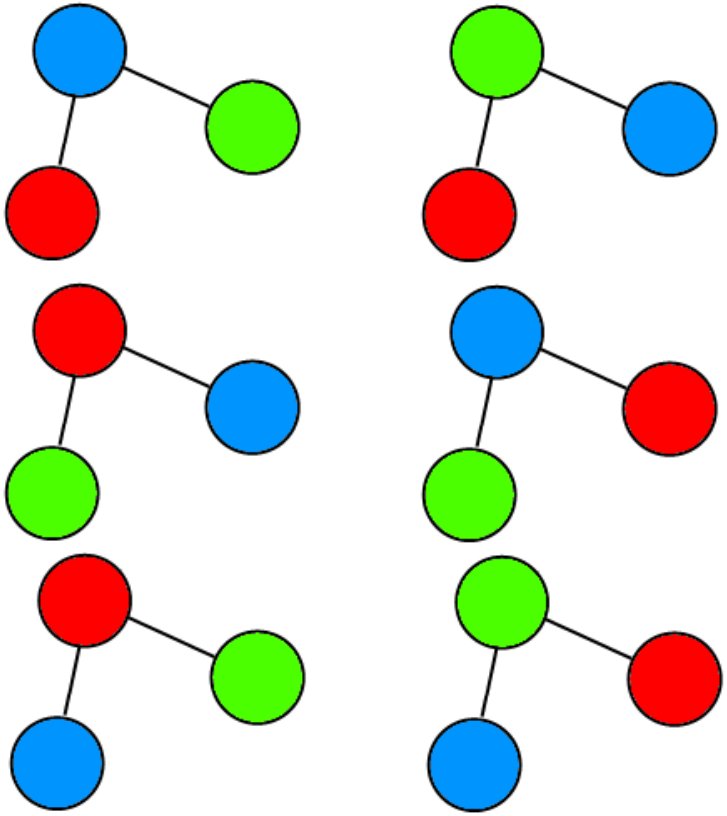
k = 3

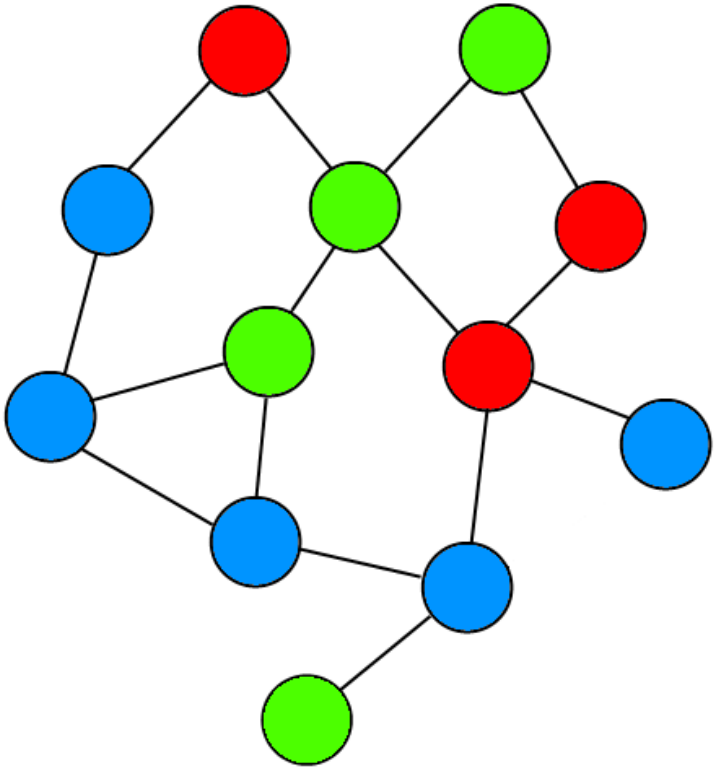
Randomly “color” vertices of graph



Color-coding approximation strategy

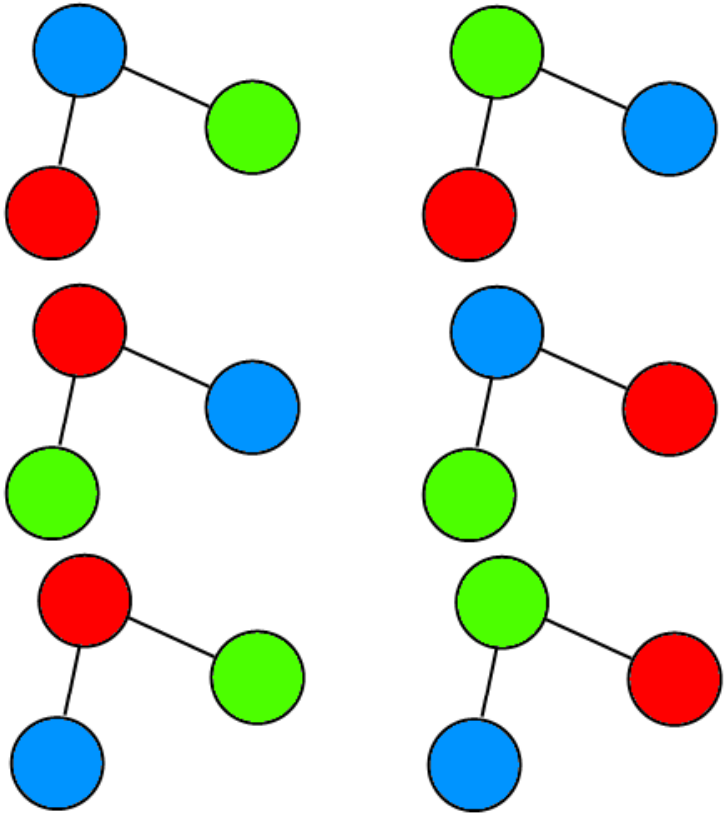
Possible colorful embeddings



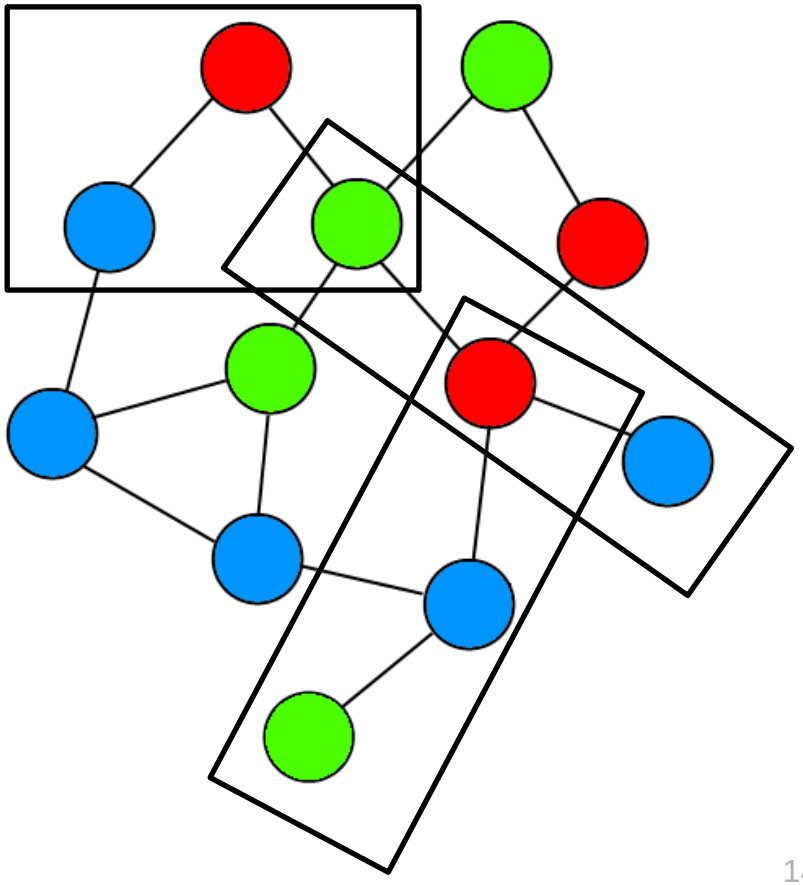


Color-coding approximation strategy

Possible colorful embeddings

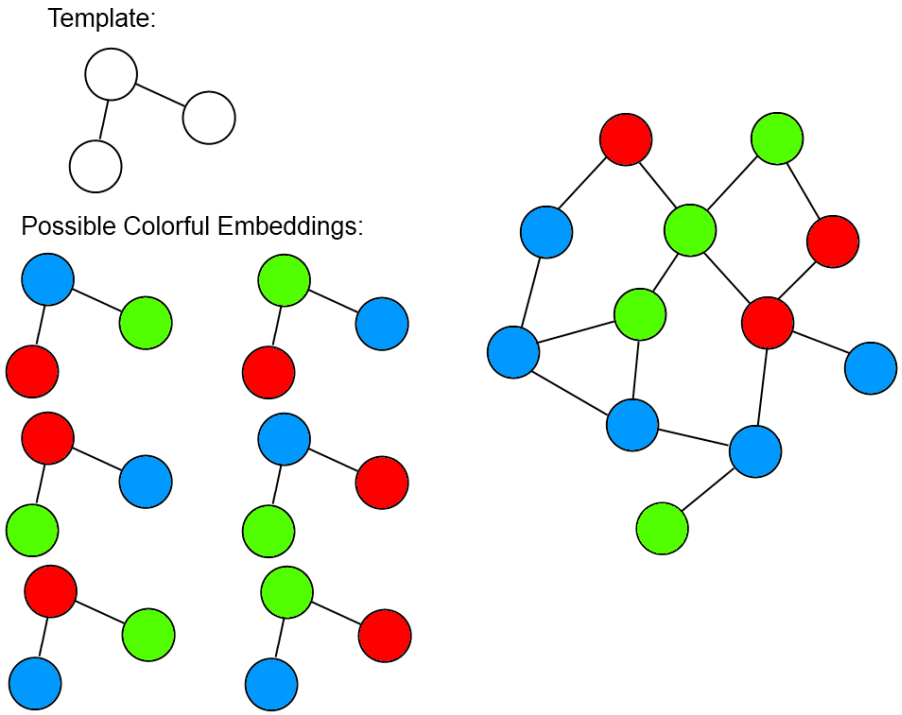


Identify colorful embeddings



Color-coding approximation strategy

- Cntcolorful = 3, Probability of colorful embedding = 3!/33
- Perform multiple (~ ek) coloring iterations
- Each iteration requires O(m2k) work



FASCIA Algorithm

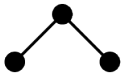
1: Partition input template T (k vertices) into subtemplates S_i using *single edge cuts*.
2: Select *Niter* to be performed
3: **for** $i = 1$ to *Niter* **do**
4: Randomly assign to each vertex v in graph G a color between 0 and $k - 1$.
5: **for all** $v \in G$ **do**
6: Use a **dynamic programming scheme** to count *colorful* non-induced occurrences of T rooted at v.
7: **end for**
8: **end for**
9: Take average of all *Niter* counts to be final count.

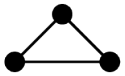
Counting step

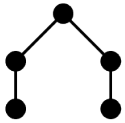
1: **for all** sub-templates S_i created from partitioning T, in reverse order they were created during partitioning **do**
2: **for all** vertices $v \in G$ **do**
3: **if** S_i consists of a single node **then**
4: Set table$[S_i][v][color\ of\ v] := 1$
5: **else**
6: S_i consists of active child a_i and passive child p_i
7: **for all** colorsets C of unique values mapped to S **do**
8: Set $count := 0$
9: **for all** $u \in N(v)$, where $N(v)$ is the neighborhood of v **do**
10: **for all** possible combinations C_a and C_p created by splitting C and mapping onto a_i and p_i **do**
11: $count$ += table$[a_i][v][C_a]$·table$[p_i][u][C_p]$
12: **end for**
13: **end for**
14: Set table$[S_i][v][C] := count$
15: **end for**
16: **end if**
17: **end for**
18: **end for**
19: $templateCount := \sum_v \sum_C$ table$[T][v][C]$

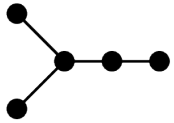
Test network families, example templates

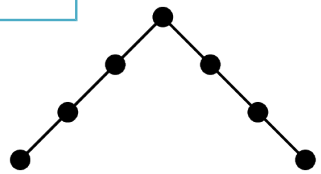
Network type	# of networks	# of edges in largest network
PINs	8	22 K
Web crawls	4	3.9 M
Social networks	6	5.4 M
Road	5	2.8 M
Collaboration	6	1.05 M
Large soc. net (Orkut)	1	117 M
Large synth. urban pop. (Portland!)	1	31 M

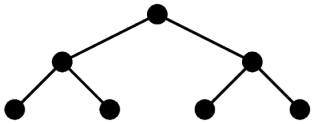

U3-1

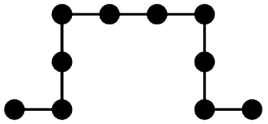

U3-2

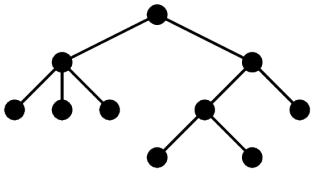

U5-1

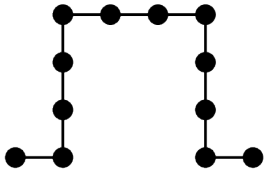

U5-2

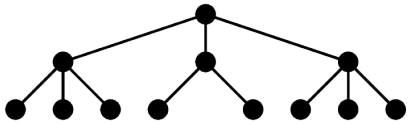

U7-1


U7-2

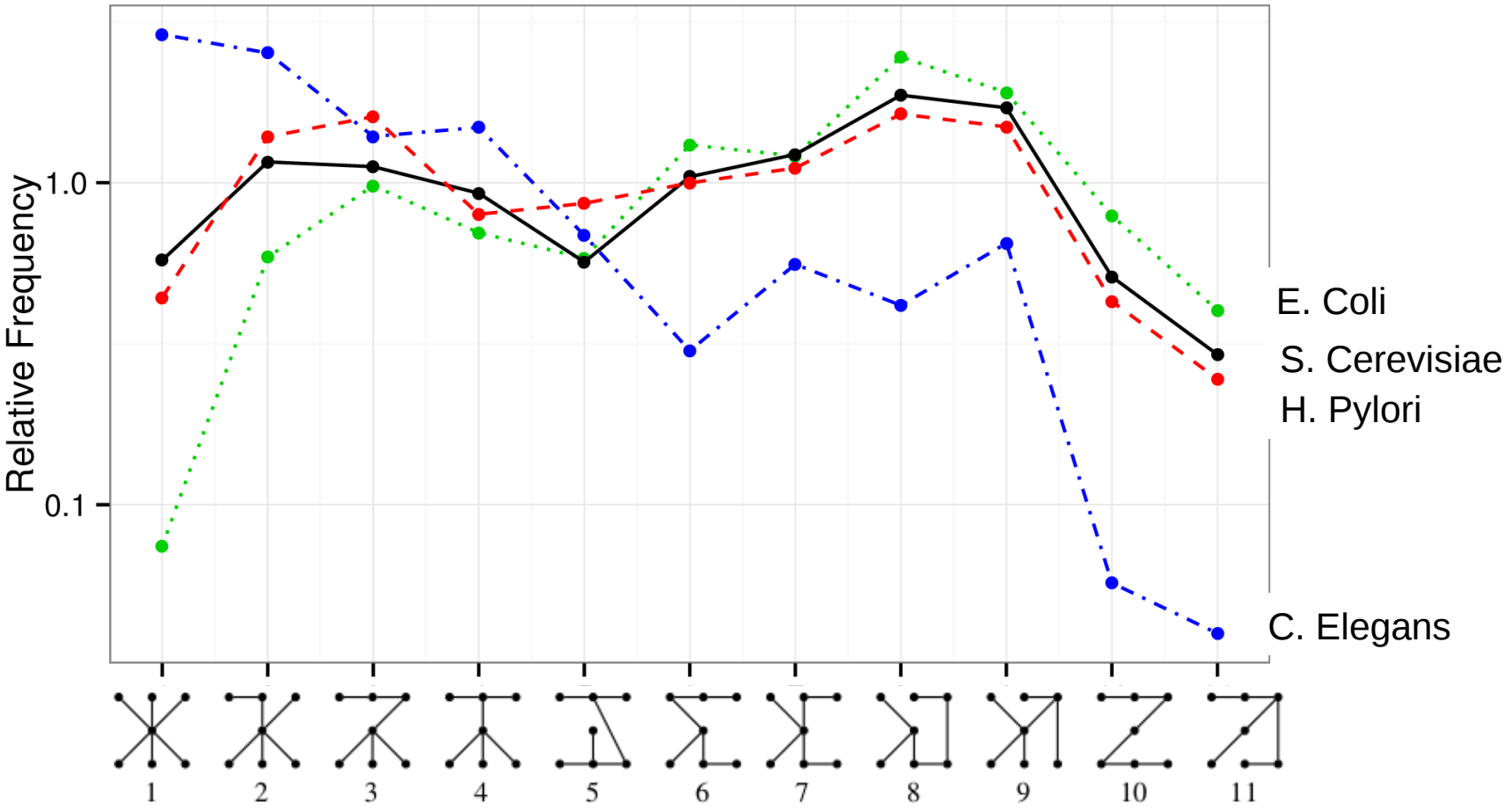

U10-1


U10-2


U12-1

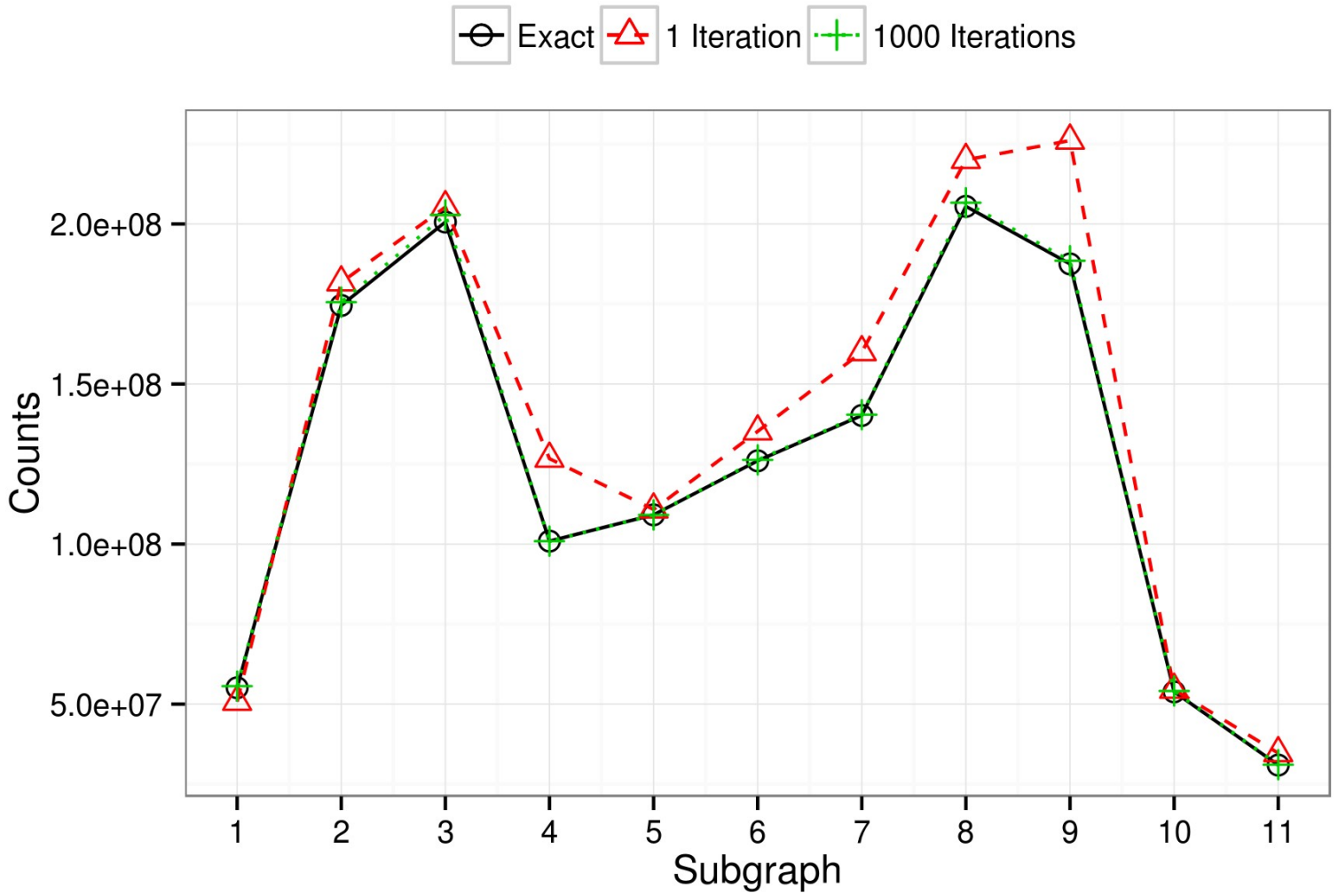

U12-2

Motifs: frequently-occurring subgraphs of certain size and structure



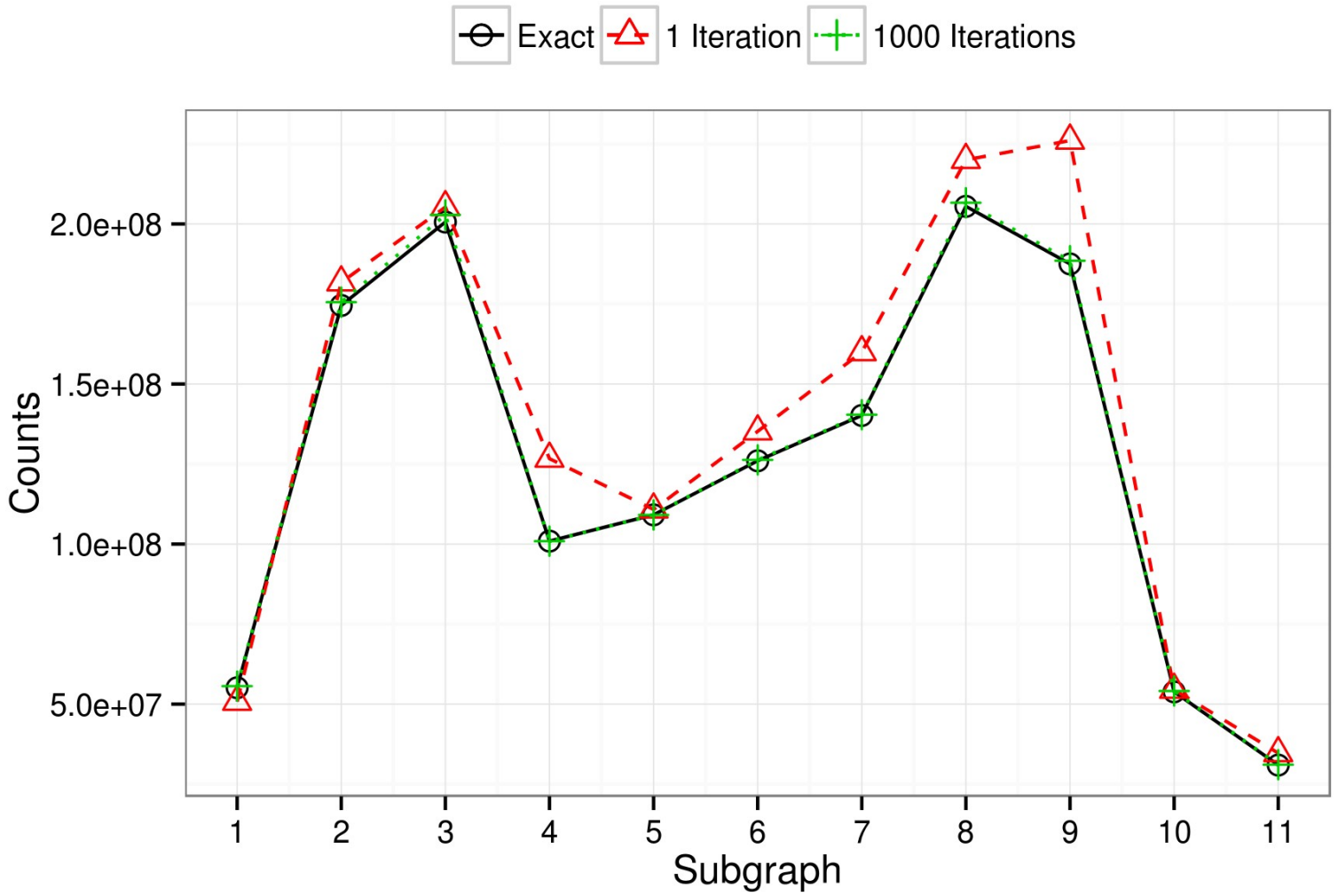
Error

H. Pylori, Subgraphs of size 7



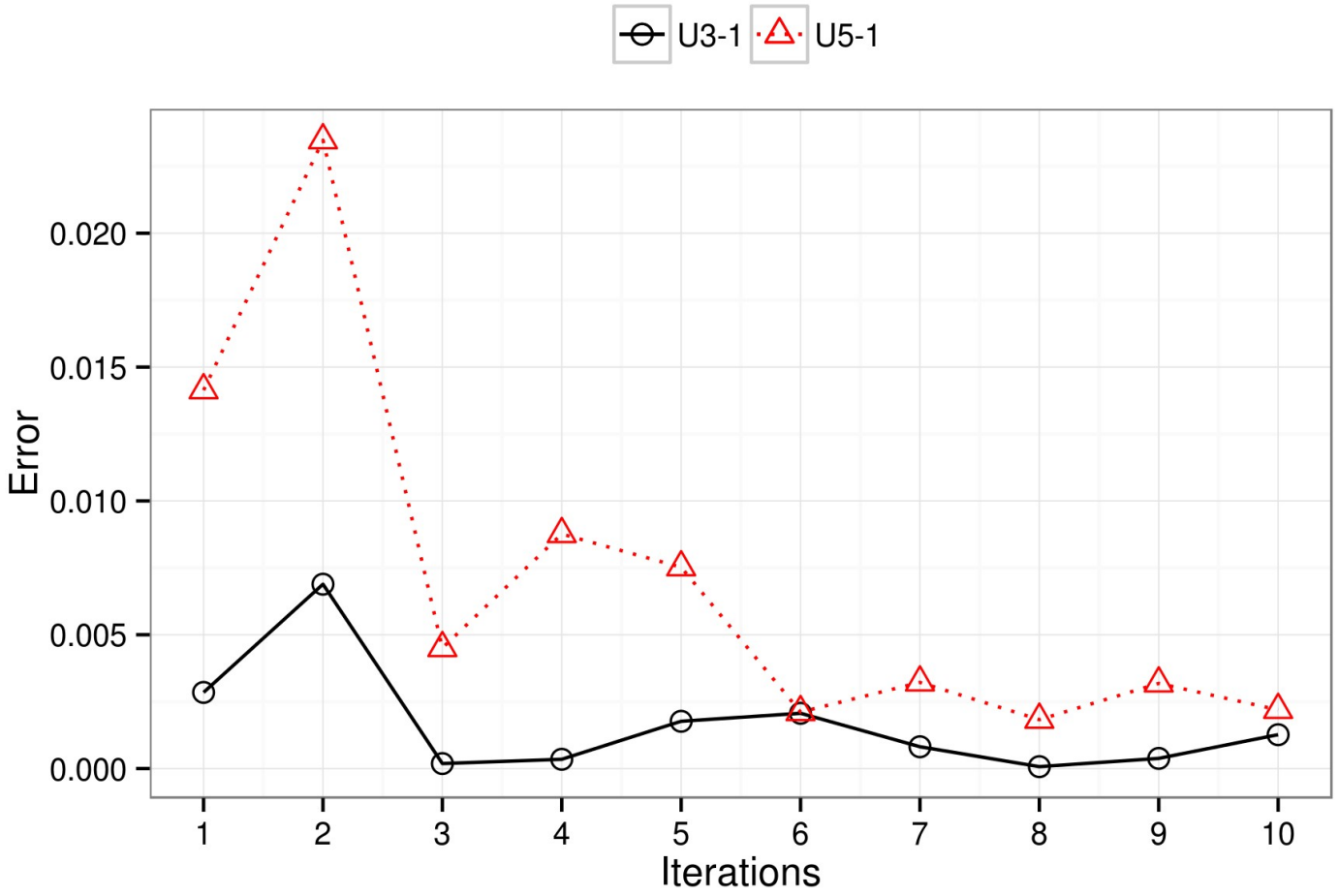
Error

H. Pylori, Subgraphs of size 7



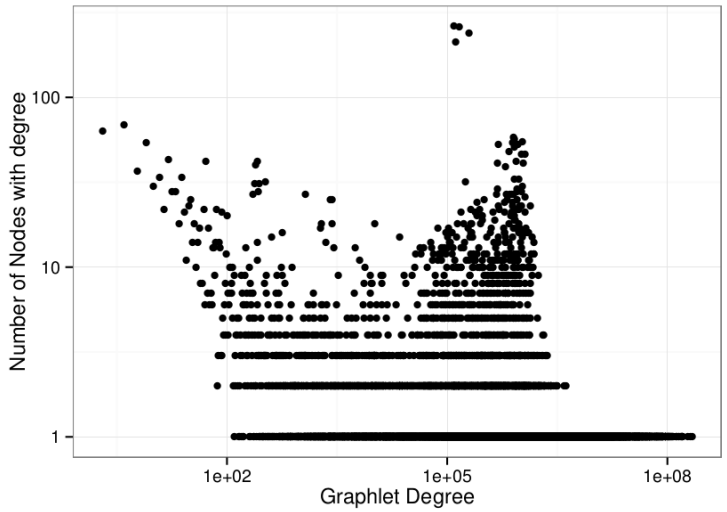
Error

Enron email network

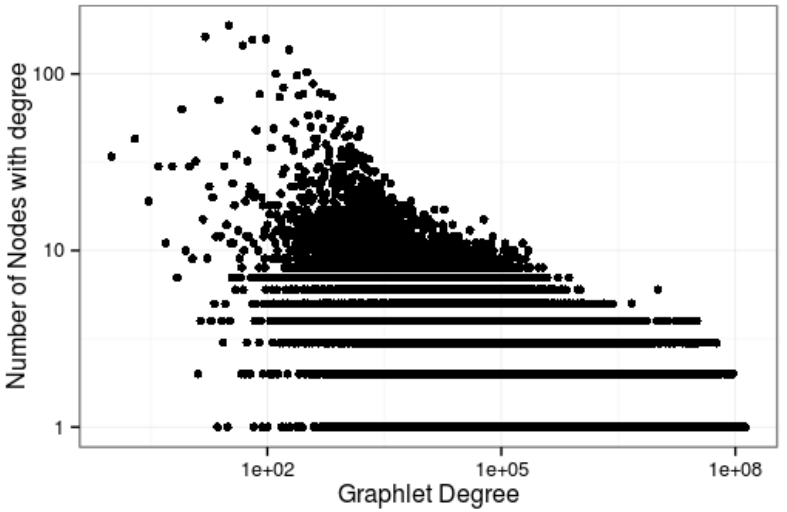


Graphlet degree distributions

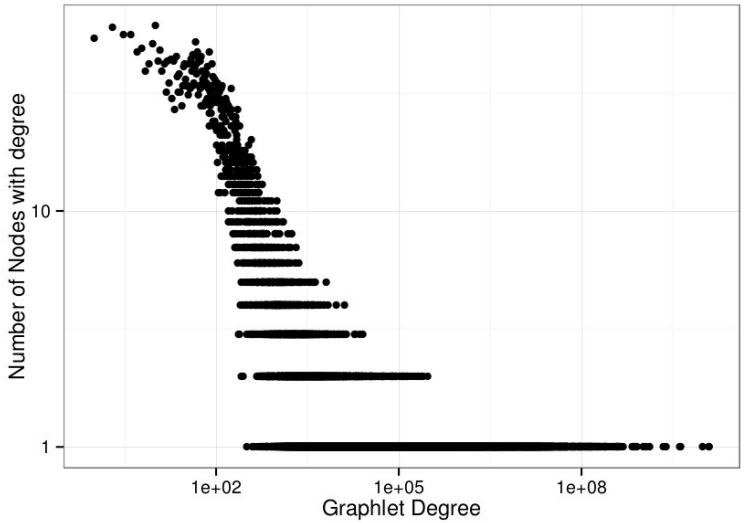
Enron



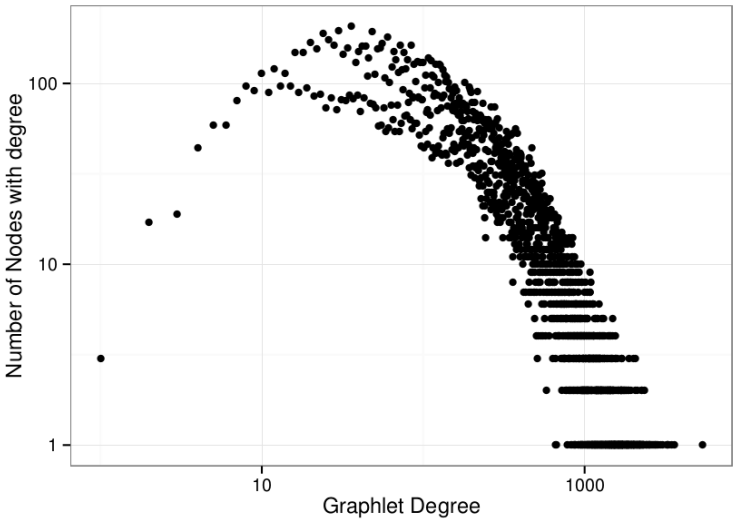
Portland



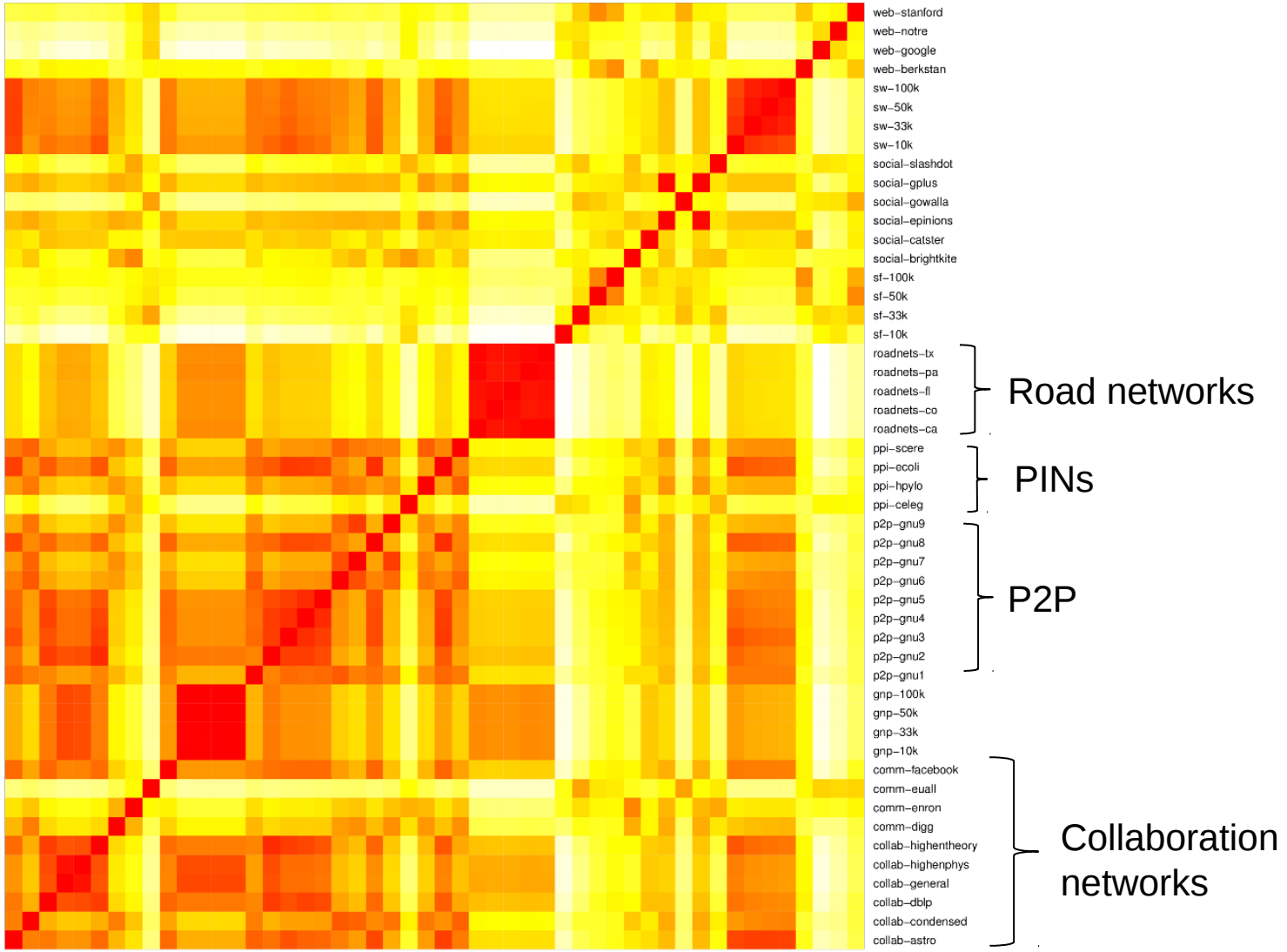
Slashdot



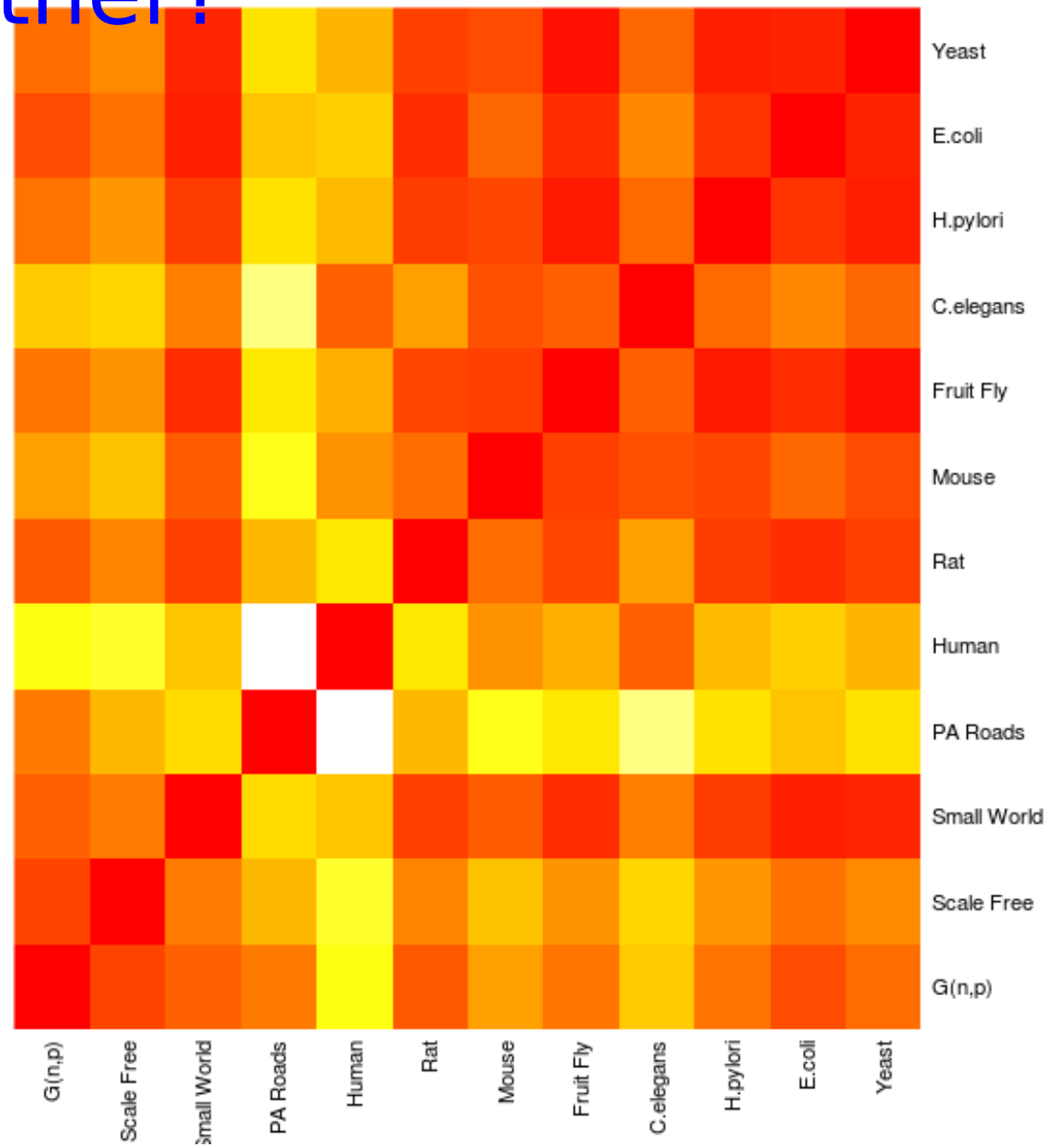
G(n,p) random



Graphlet frequency distribution agreement scores heatmap



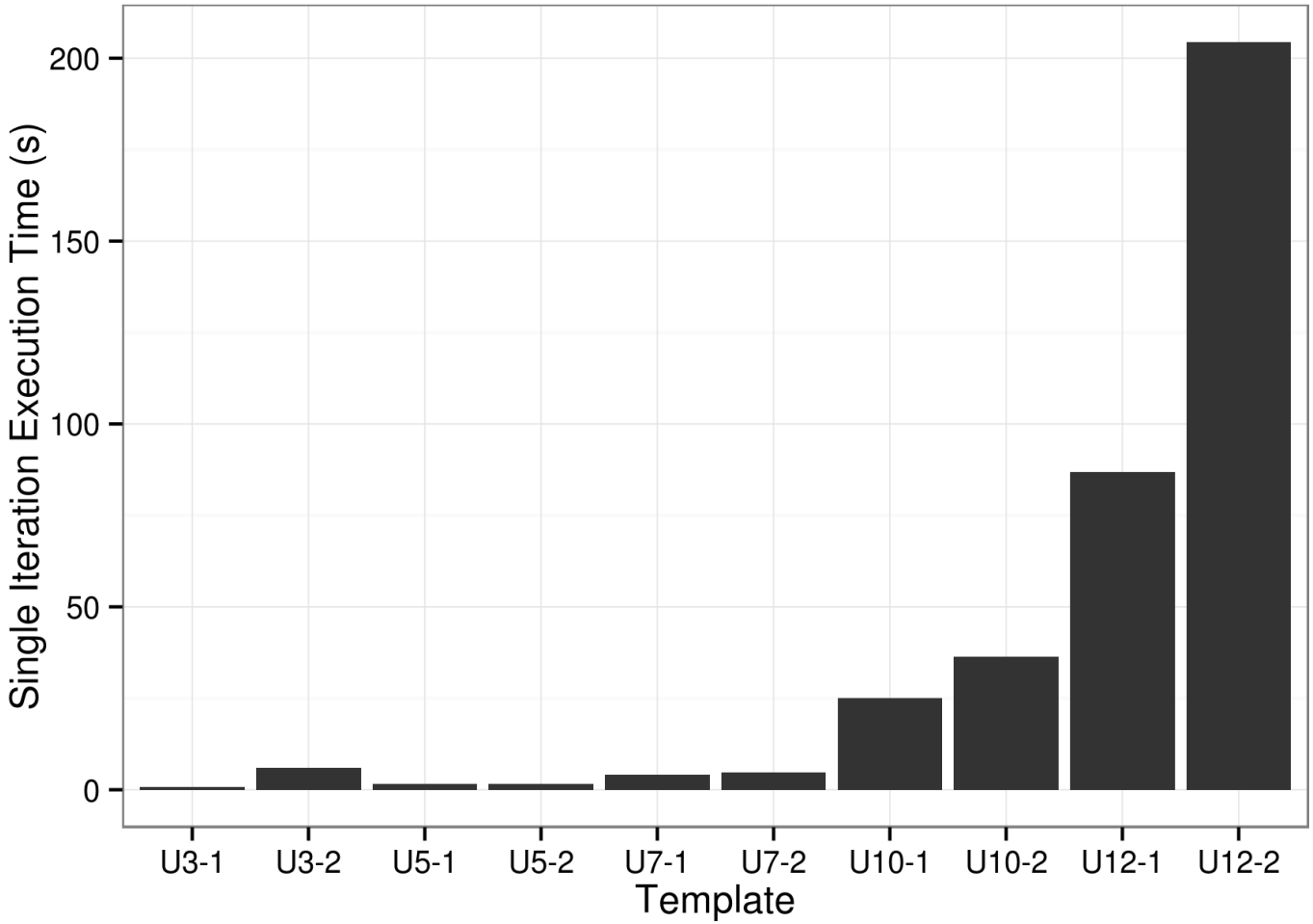
How similar are PINs to each other?



Execution times for various template sizes

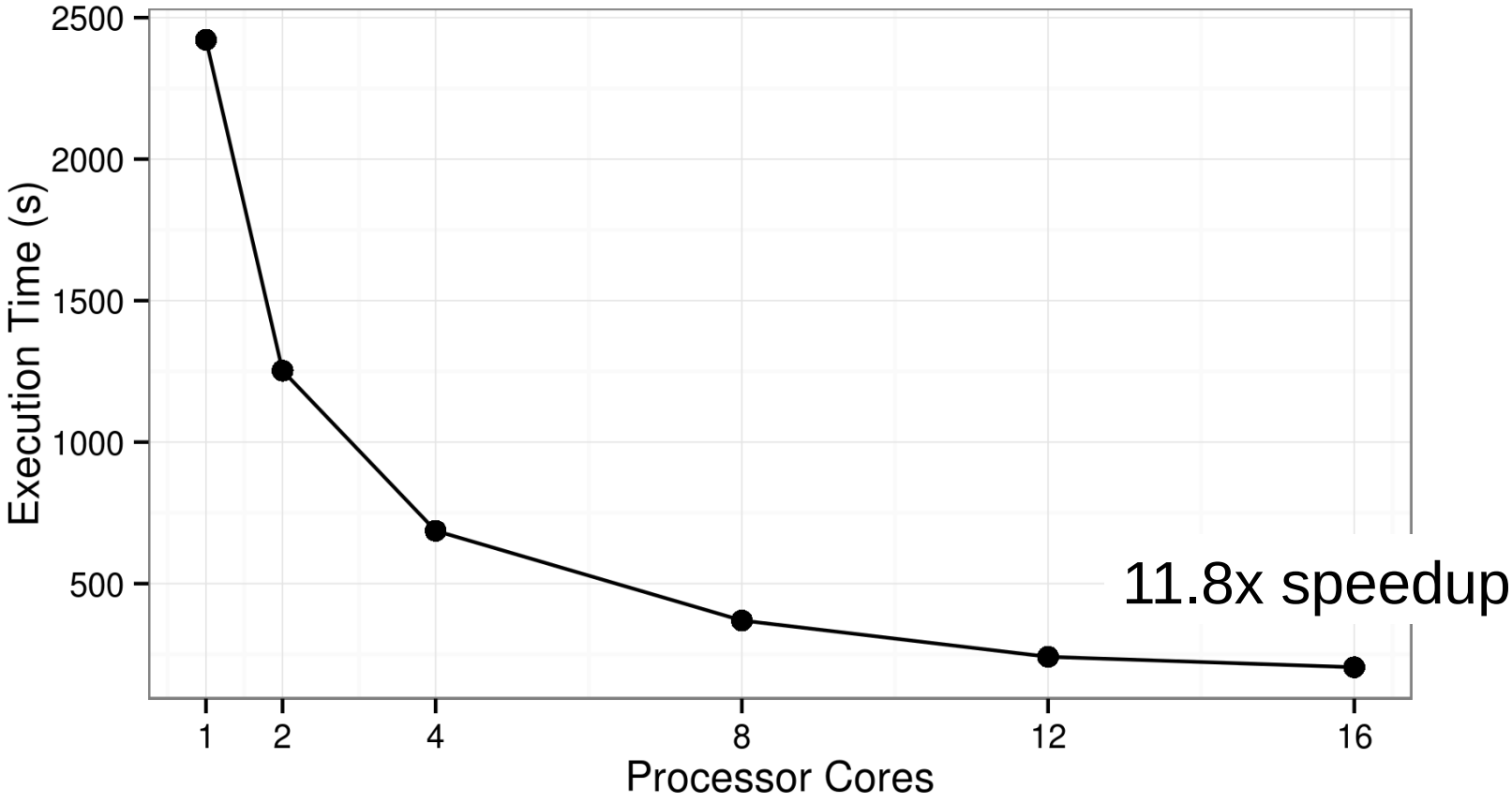
Portland network (31M edges)

Single node performance (Intel Sandy Bridge server, 16 cores)



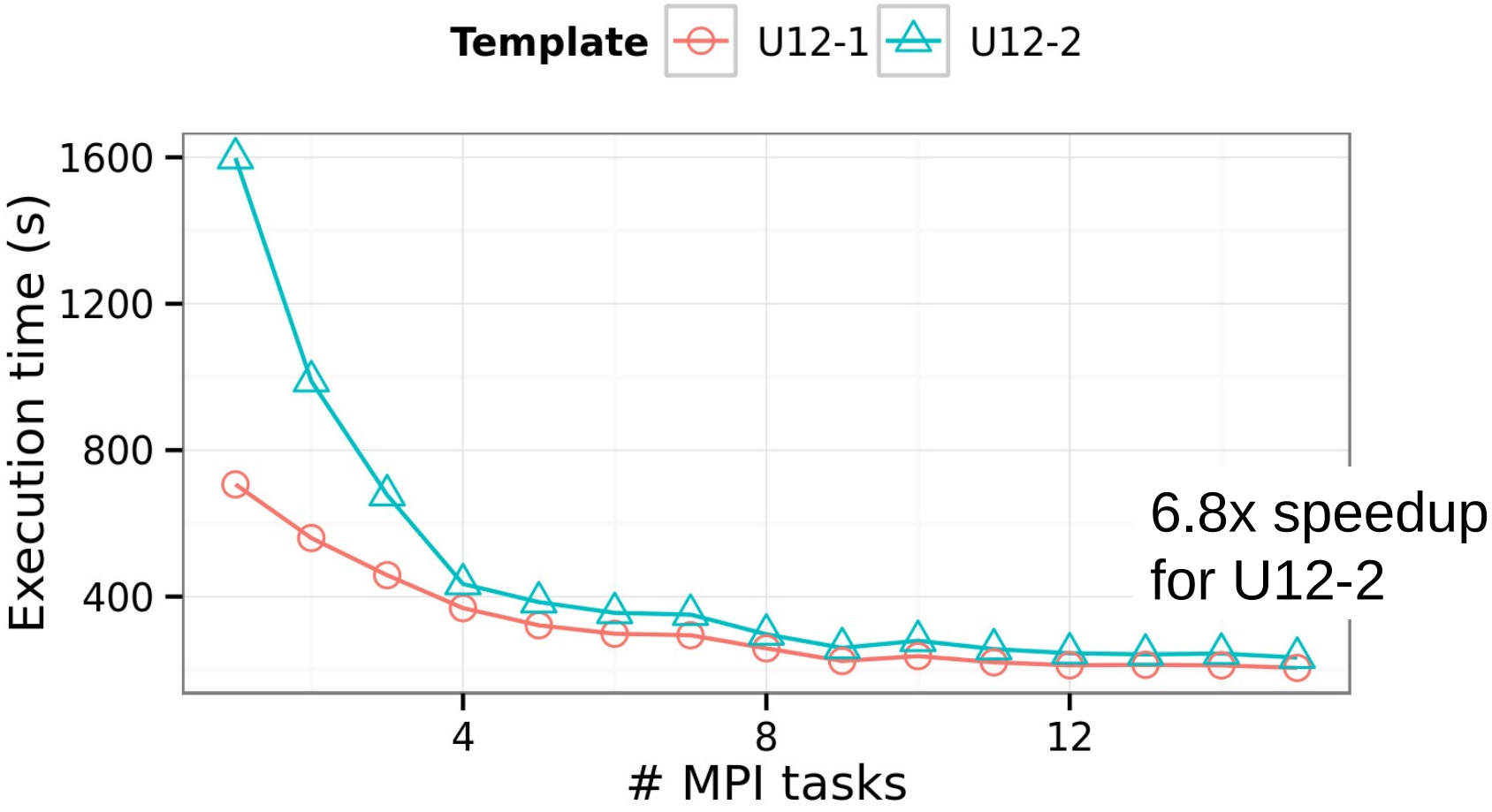
Shared-memory strong scaling

Portland network (31M edges), U12-2 template
Single node performance (Intel Sandy Bridge server, 16 cores)
1 color-coding iteration



Multi-node strong scaling

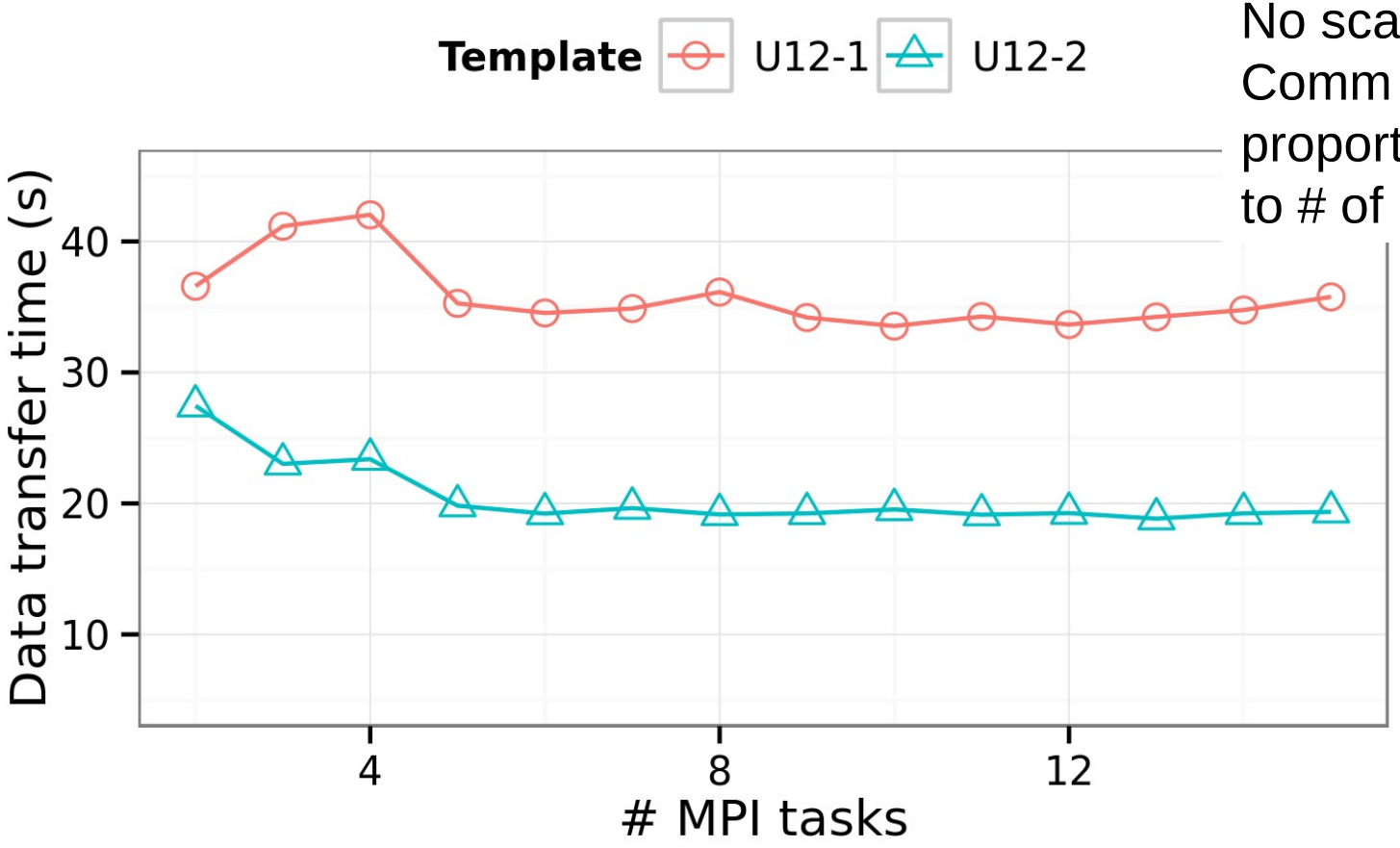
Orkut network (117M edges)
Performance on an Intel Sandy Bridge cluster (1-15 nodes)



Multi-node strong scaling (communication time)

Orkut network (117M edges)

Performance on an Intel Sandy Bridge cluster (1-15 nodes)



No scaling:
Comm Volume
proportional
to # of MPI tasks!

FASCIA Summary and Current Work

- FASCIA: Color-coding parallel implementation applied to count subgraphs
- Current work: FastPath, color-coding applied to enumerate simple short paths
- FASCIA performance optimization
  - Graph coarsening
  - Reduce memory utilization
  - Fix the (computation) load imbalance at high concurrencies with better graph partitioning
  - Hide inter-node communication, non-blocking collectives
- Source code: Will be up at fascia-psu.sourceforge.net

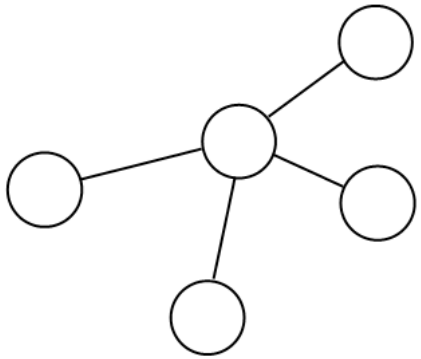
Thank you!

- Questions?

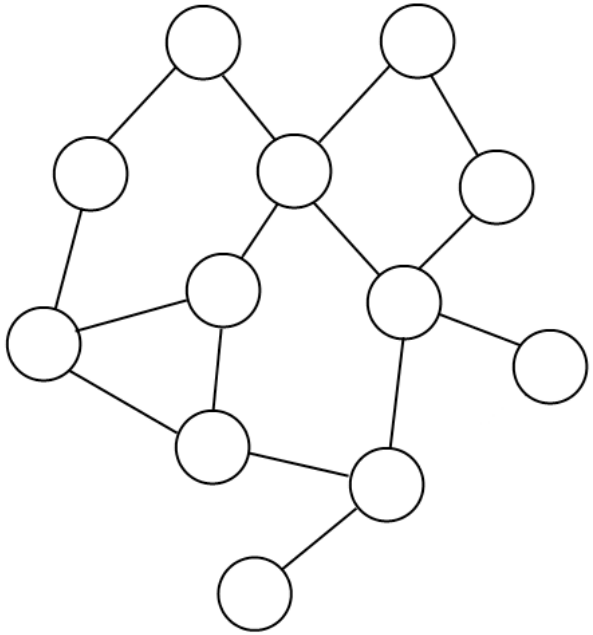
- madduri@cse.psu.edu
- http://www.cse.psu.edu/~madduri

Backup slides

Subgraph counting

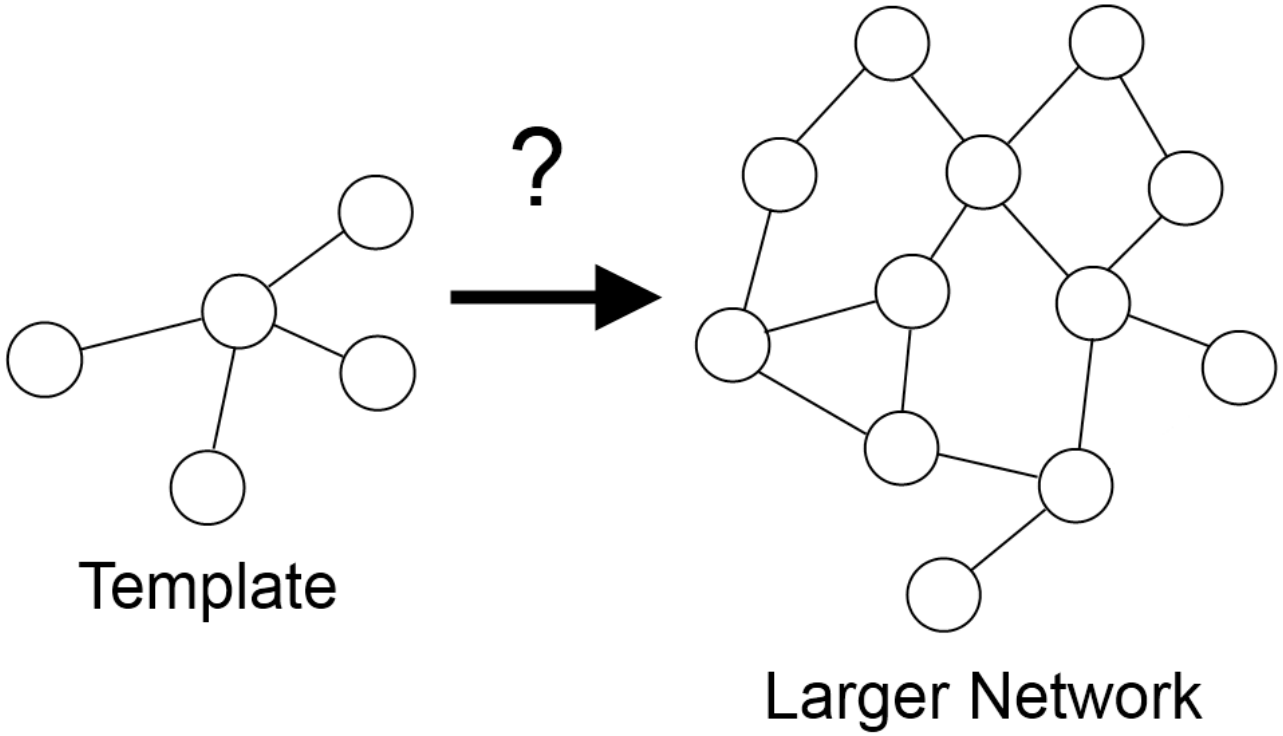


Template

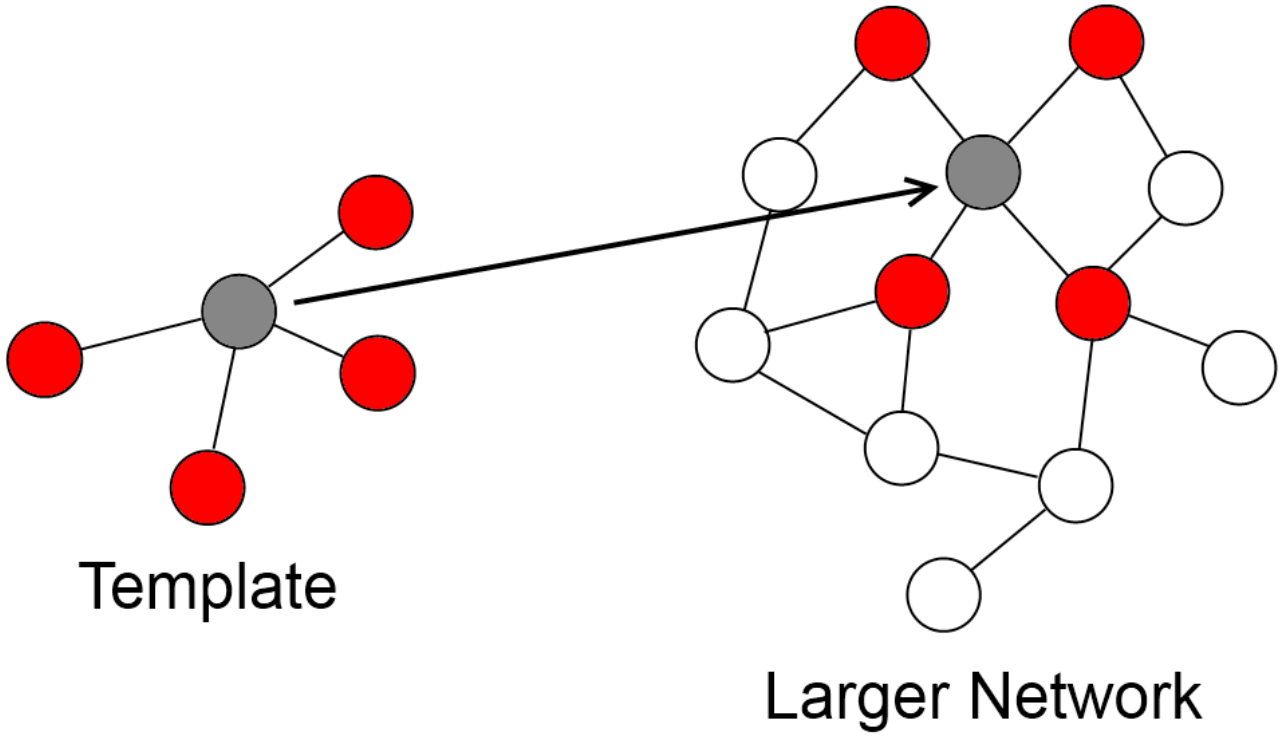


Larger Network

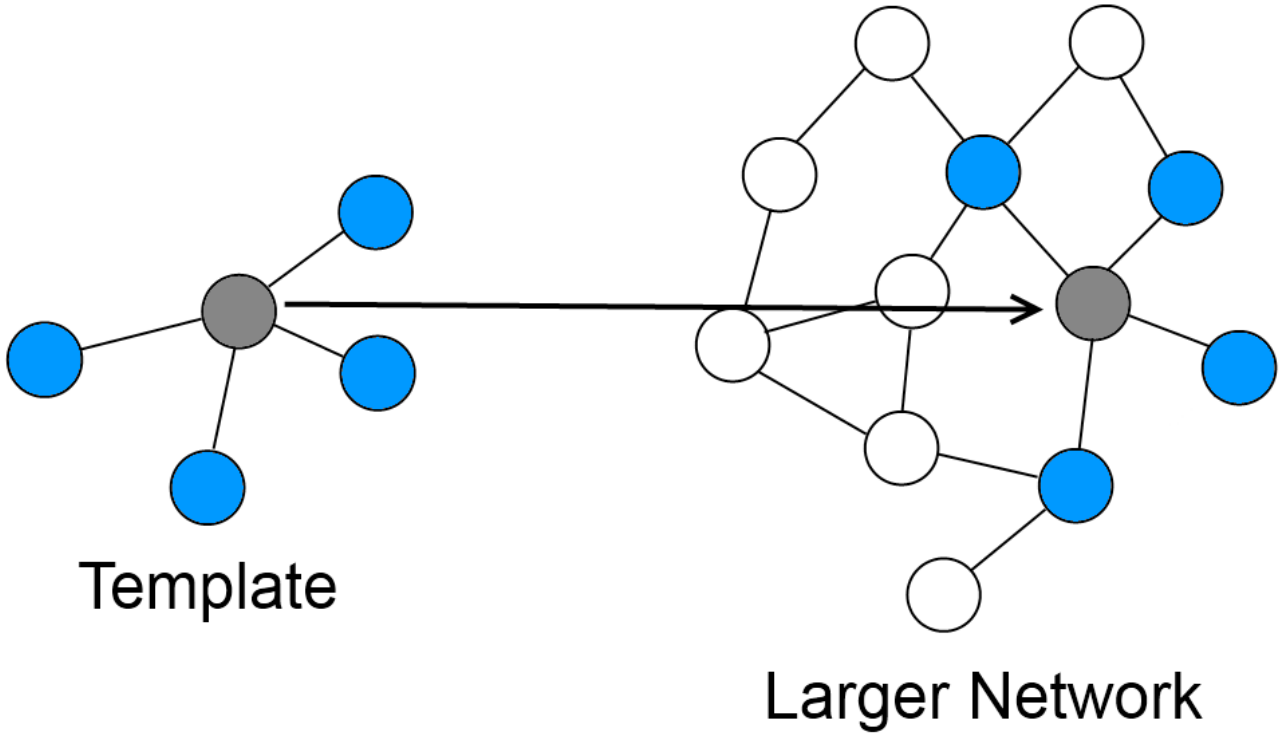
Subgraph counting



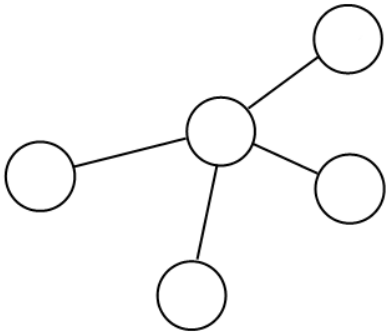
Subgraph counting

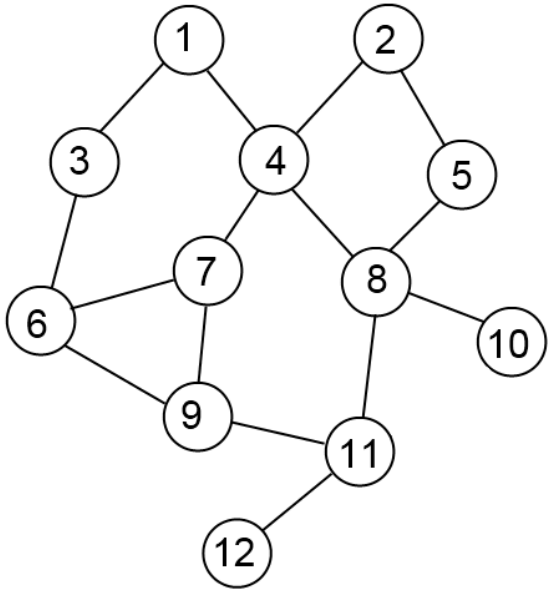


Subgraph counting

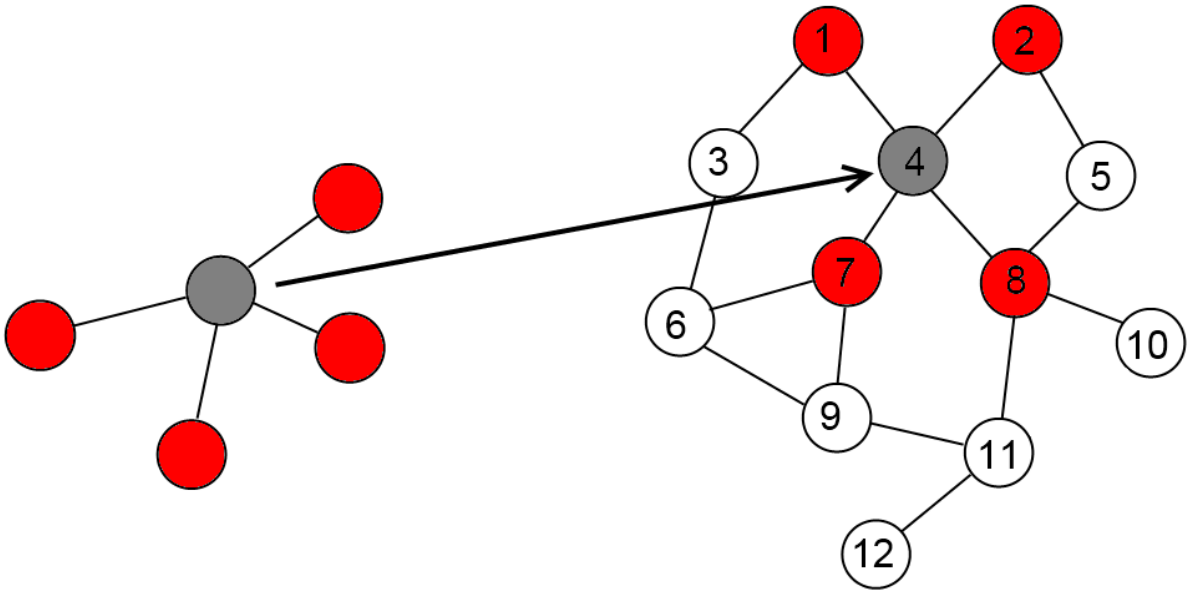


Subgraph enumeration



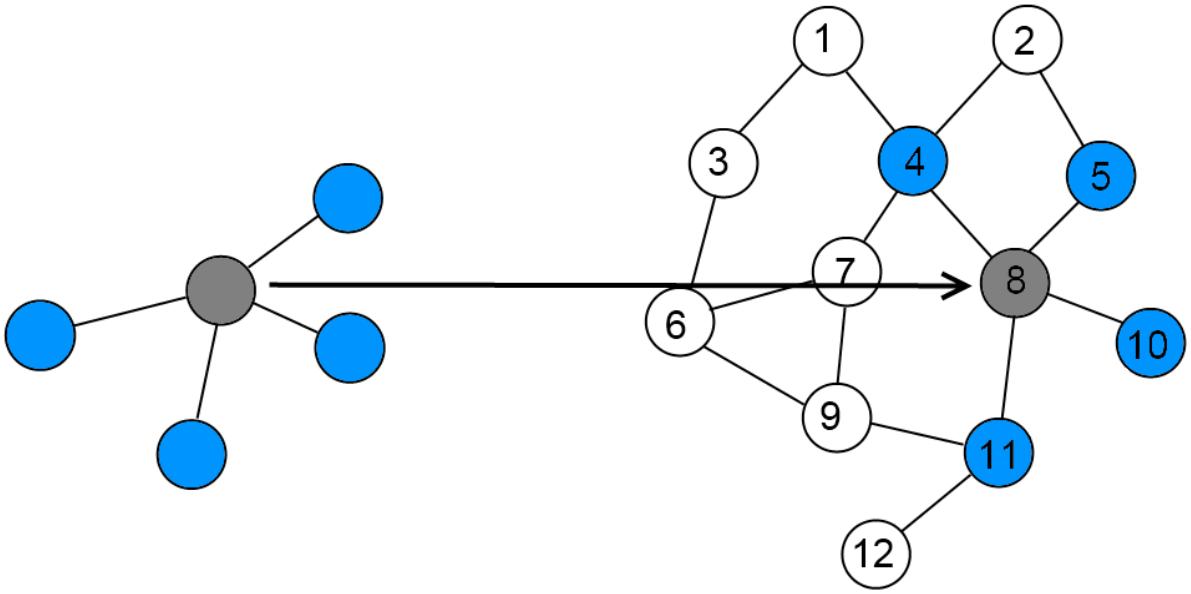


Subgraph enumeration



Mapped to vertices:
4 1 2 7 8

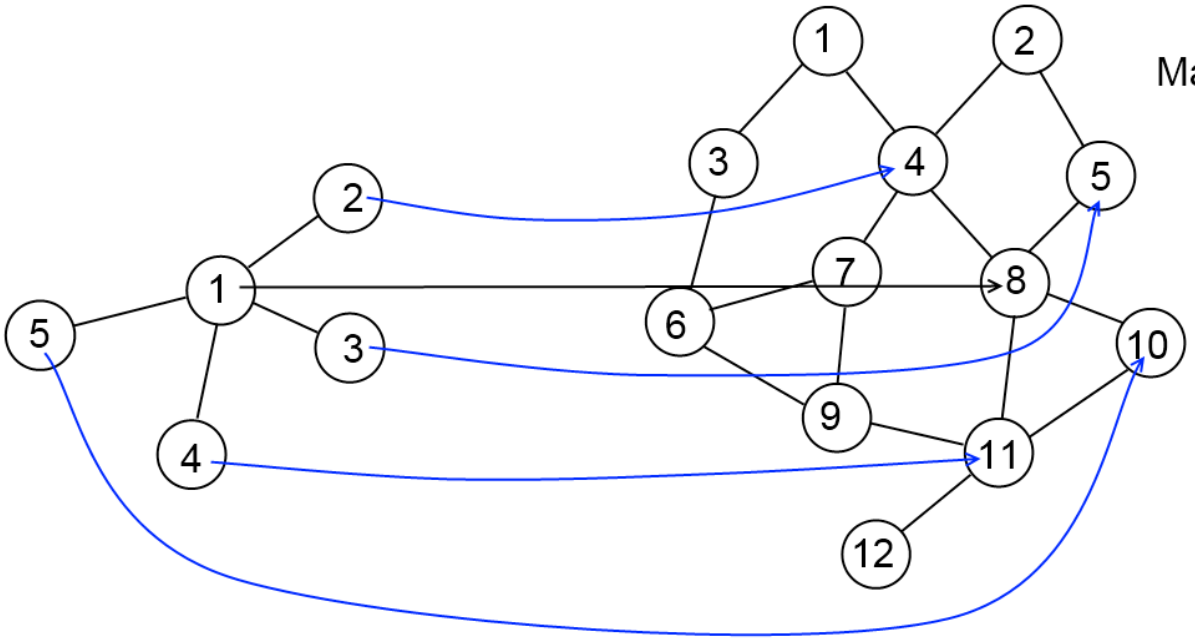
Subgraph enumeration



Mapped to vertices:

4	1	2	7	8
8	4	5	11	10

Subgraph enumeration



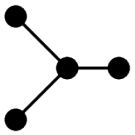
Subgraph vertices:

1	2	3	4	5

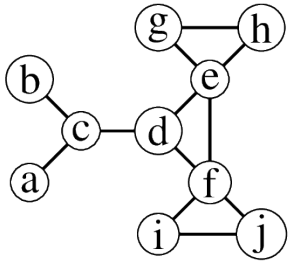
Mapped to Graph vertices:

4	1	2	8	7
8	4	5	11	10

Induced vs. Non-induced subgraphs



G_T



G

Motivation: Why fast algorithms for subgraph counting?

- Important in bioinformatics, chemoinformatics, social network analysis, communication network analysis, etc.
- Forms basis of more complex analysis
  - Motif finding
  - Graphlet frequency distance (GFD)
  - Graphlet degree distributions (GDD)
  - Graphlet degree signatures (GDS)
- Exact counting and enumeration on large networks is very compute-intensive, O(nk) work complexity for naïve algorithm

GFD

- Numerically compare occurrence frequency to other networks

$$S_i(G) = -\log\left(\frac{C_i(G)}{\sum_{i=1}^{n} C_i(G)}\right)$$

$$D(G,H) = \sum_{i=1}^{n} |S_i(G) - S_i(H)|$$